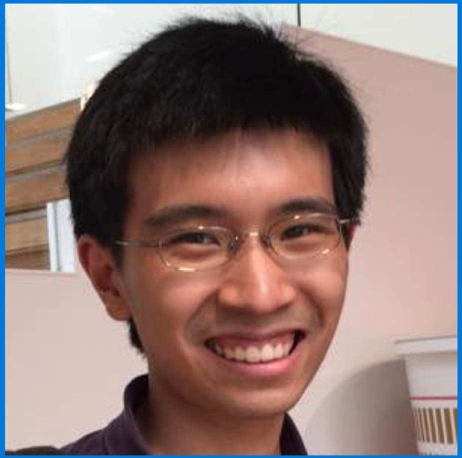
Quantum Algorithms for Composed Functions With Shared Inputs



Mark Bun
Simons Institute

Robin Kothari
Microsoft Research

Justin Thaler
Georgetown University

Introduction

Query complexity

Let $f: \{0,1\}^n \to \{0,1\}$ be a function and $x \in \{0,1\}^n$ be an input to f.

$$x = \begin{array}{|c|c|c|c|c|}\hline x_1 & x_2 & x_3 & \quad\cdots\quad & x_n \\ \hline\end{array}$$

Goal: Compute $f(x)$ by reading as few bits of x as possible.

Equivalently, compute $f(x)$ using a circuit/algorithm with the least number of uses of this oracle:

$$i \longrightarrow \boxed{O_x} \longrightarrow x_i$$

In the quantum setting, we have this oracle:

$$|i\rangle|b\rangle \longrightarrow \boxed{O_x} \longrightarrow |i\rangle|b \oplus x_i\rangle$$

Why query complexity?

Quantum algorithmic motivation

- Query algorithms often can be made time-efficient, while the abstraction of query complexity often gets rid of unnecessary details.
- Most quantum algorithms are naturally phrased as query algorithms. E.g., Shor, Grover, Hidden Subgroup, Linear systems (HHL), etc.

Classical algorithmic motivation

- Functions with quantum query complexity $< d$ can be agnostically PAC learned in time $\sim n^d$.

Classical complexity motivation

- If all circuits in a class C have quantum query complexity o(n), then no circuit from C can correctly compute IP2 on a $\frac{1}{2} + 1/\mathrm{poly}(n)$ fraction of inputs.

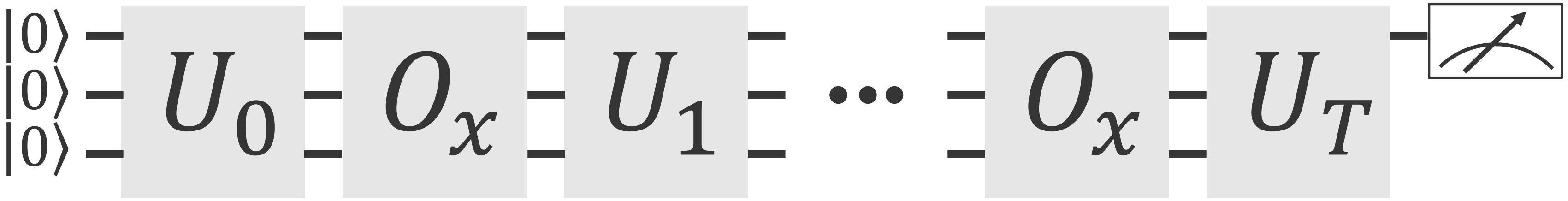
Other applications

- Oracle separations between classes, lower bounds on restricted models, upper and lower bounds in communication complexity, data structures, etc.

Quantum query complexity

Quantum query complexity: Minimum number of uses of O_x in a quantum algorithm that for every input x, outputs $f(x)$ with error $\leq 1/3$.

$Q(f)$



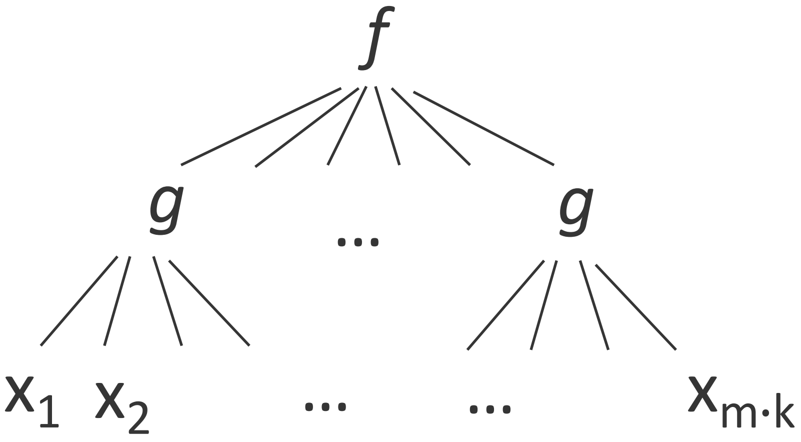
Example: Let $\mathrm{OR}_n(x) = \bigvee_{i=1}^{n} x_i$ and $\mathrm{AND}_n(x) = \bigwedge_{i=1}^{n} x_i$.

Then $Q(\mathrm{OR}_n) = Q(\mathrm{AND}_n) = \Theta(\sqrt{n})$ [Grover96, Bennett-Bernstein-Brassard-Vazirani97]

Classically, we need $\Theta(n)$ queries for both problems.

Query Complexity of Block-Composed Functions

- If $f: \{0,1\}^m \to \{0,1\}$ and $g: \{0,1\}^k \to \{0,1\}$, the block composition $f \circ g: \{0,1\}^{m \cdot k} \to \{0,1\}$ is defined via $f \circ g = f(g, \ldots, g)$, with each copy of g on a disjoint set of variables.

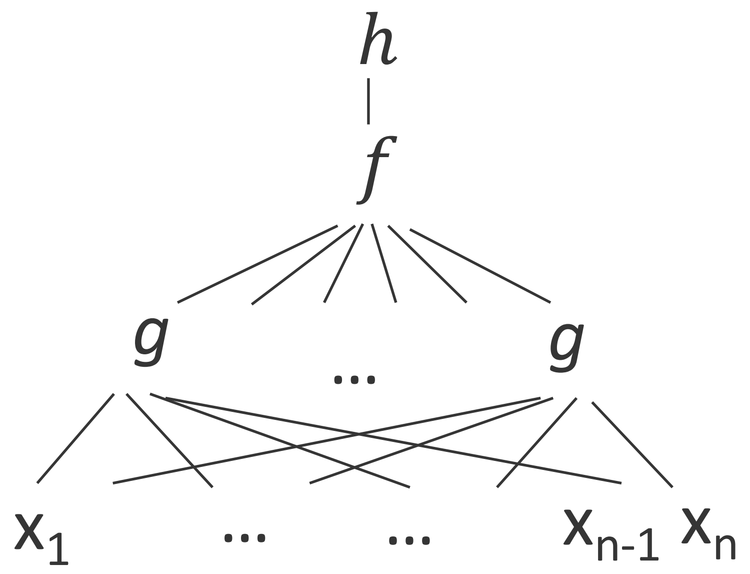


- Fact: $Q(f \circ g) = \Theta\big(Q(f) \cdot Q(g)\big)$ [HLS 2007, Reichardt 2010].
- **Tomorrow:** Troy Lee's talk will discuss whether the analogous statement holds for randomized query complexity.

Main Composition Theorem

Query Complexity of Shared-Input Compositions

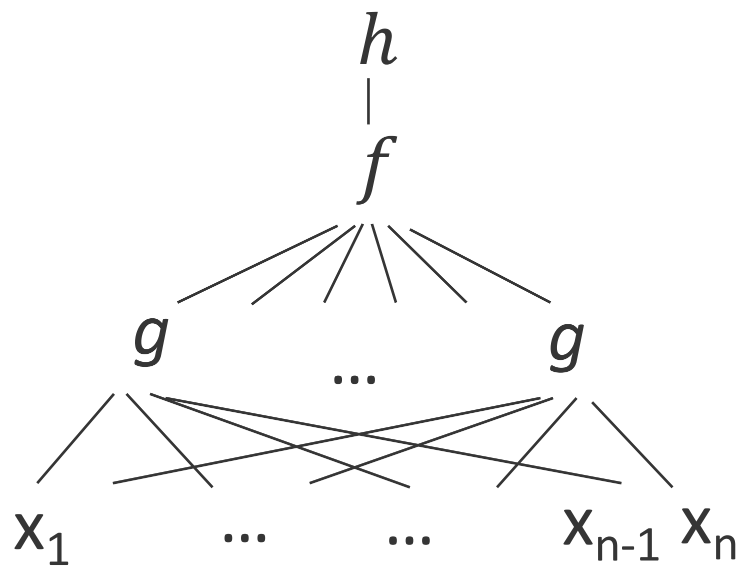
- We are interested in shared-input compositions.



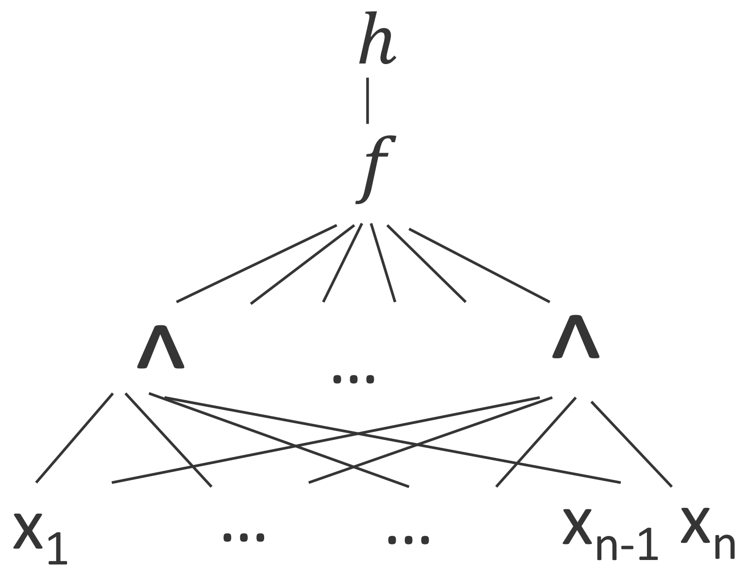
- A trivial upper bound for $Q(h)$ is: $Q(h) \leq O(Q(f) \cdot Q(g))$.
- Is it always possible to leverage the sharing of inputs to improve this upper bound?

Query Complexity of Shared-Input Compositions

- We are interested in shared-input compositions.

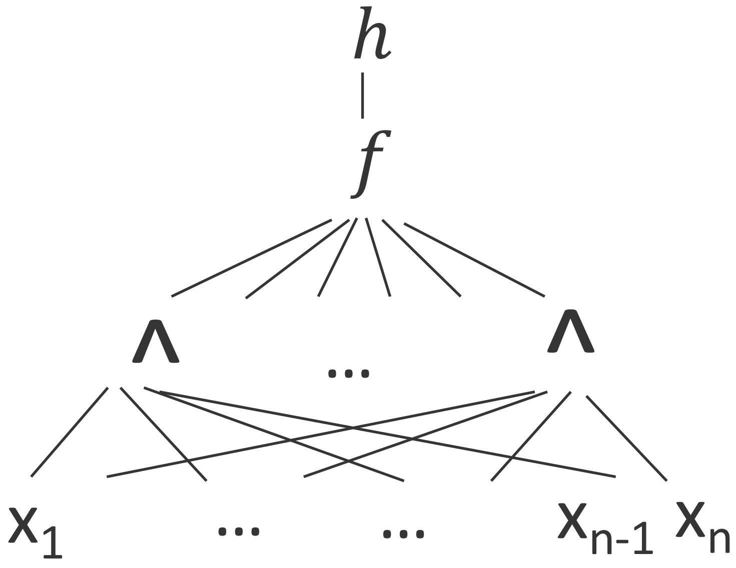


- A trivial upper bound for $Q(h)$ is: $Q(h) \le O(Q(f) \cdot Q(g))$.
- Is it always possible to leverage the sharing of inputs to improve this upper bound?
- We show the answer is **YES** whenever g=AND, and $Q(f) \ll n$.



Main Theorem: $Q(h) \le O\left(\sqrt{Q(f)\cdot n}\cdot polylog(n)\right)$.

- Without leveraging input sharing, the best upper bound for $Q(h)$ is: $Q(h) \le O(Q(f)\cdot\sqrt{n})$.



Main Theorem: $Q(h) \le O\left(\sqrt{Q(f) \cdot n} \cdot polylog(n)\right)$.

- Without leveraging input sharing, the best upper bound for $Q(h)$ is: $Q(h) \le O(Q(f) \cdot \sqrt{n})$.
- Also show an analog for **approximate degree**: $\mathrm{adeg}(h) \le \tilde{O}\left(\sqrt{\mathrm{adeg}(f) \cdot n}\right)$.
- Both results are tight for some functions, e.g., for $h = PARITY_t \circ AND_{\frac{n}{t}}$.

Idea of the Quantum Algorithm for h

- Query input bits with the goal of “killing” high fan-in AND gates.
  - i.e., if we query bit x_i and learn $x_i = 0$, then all AND gates involving x_i have their value fixed to 0, so we can effectively delete them.

Idea of the Quantum Algorithm for h

- Query input bits with the goal of "killing" high fan-in AND gates.
  - i.e., if we query bit x_i and learn $x_i = 0$, then all AND gates involving x_i have their value fixed to 0, so we can effectively delete them.

- Let h' denote h with all queried bits x_i restricted to their queried values.
- If all surviving AND gates have fan-in at most $n/Q(f)$, then the trivial upper bound yields:

$$Q(h') \leq Q(f) \cdot \sqrt{\frac{n}{Q(f)}} = \sqrt{n \cdot Q(f)}.$$

Idea of the Quantum Algorithm for h

- Remaining challenge: Figure out how to reduce the fan-in of all surviving AND gates to $n/Q(f)$, using only $\sqrt{n \cdot Q(f)}$ queries.
  - To accomplish this, we iteratively Grover search for an x_i such that:
    - $x_i = 0$ and
    - x_i is connected to “many” surviving high-fan-in AND gates.

Implications of the Composition Theorem

Quantum Query Algorithms for Linear-Size AC^0

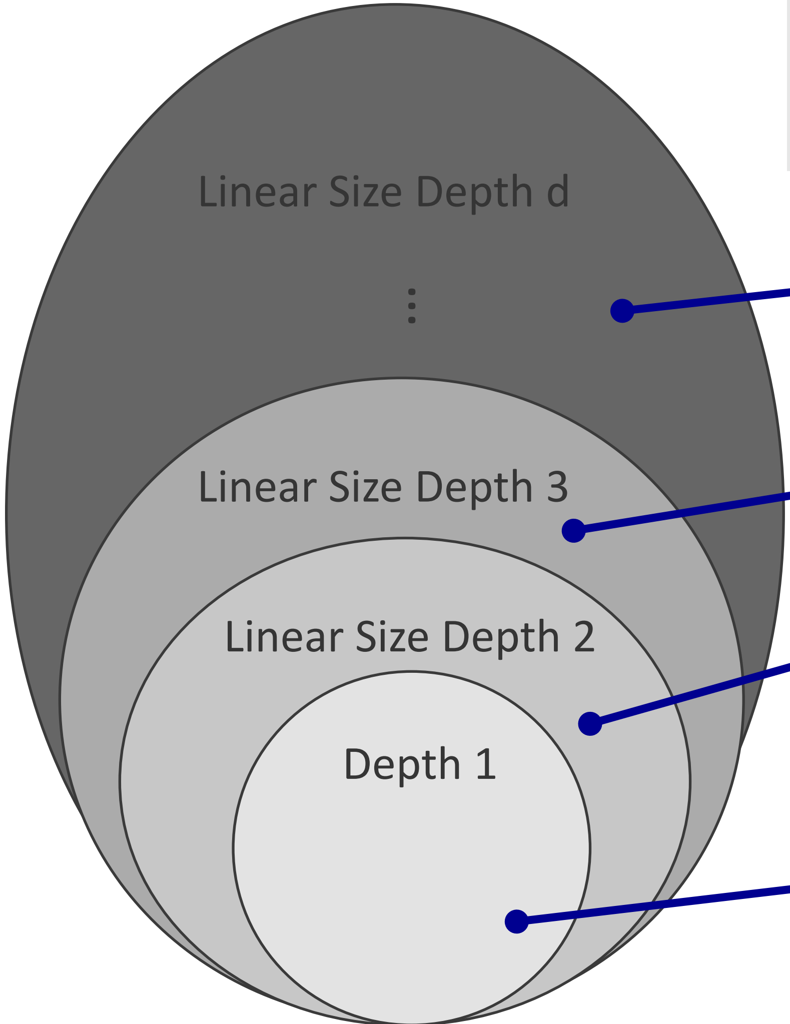
- Recursively applying our Main Theorem shows that any linear size, depth-d (i.e., LC^0_d) circuit f satisfies $Q(f) = \widetilde{O}(n^{1-2^{-d}})$.

Main Theorem: $Q(h) \leq O\left(\sqrt{Q(f) \cdot n} \cdot polylog(n)\right)$.

Quantum Query Algorithms for Linear-Size AC^0

- Recursively applying our Main Theorem shows that any linear size, depth-d (i.e., LC$_d^0$) circuit f satisfies Q$(f) = \widetilde{O}(n^{1-2^{-d}})$.

Main Theorem: $Q(h) \le O\left(\sqrt{Q(f) \cdot n} \cdot polylog(n)\right)$.



$Q(f) = \widetilde{\mathrm{O}}\left(n^{1-2^{-d}}\right)$ [new]

$Q(f) = \widetilde{\mathrm{O}}\left(n^{7/8}\right)$ [new]

$Q(f) = \widetilde{\mathrm{O}}\left(n^{3/4}\right)$ [Childs, Kothari, Kimmel 2012]

$Q(\mathrm{OR}_n) = \mathrm{O}(\sqrt{n})$ [Grover96, BBBV97]

Quantum Query Algorithms for Linear-Size AC^0

- Recursively applying our Main Theorem shows that any linear size, depth-d (i.e., LC^0_d) circuit f satisfies $Q(f) = \widetilde{O}(n^{1-2^{-d}})$.
- Nearly matches known lower bound: for every d, there is a $f \in LC^0_d$ with $Q(f) = \Omega(n^{1-2^{-O(d)}})$. [Childs, Kothari, Kimmel 2012].

Quantum Query Algorithms for Linear-Size AC^0

- Recursively applying our Main Theorem shows that any linear size, depth-d (i.e., LC_d^0) circuit f satisfies $\mathrm{Q}(f) = \tilde{O}(n^{1-2^{-d}})$.
- Nearly matches known lower bound: for every d, there is a $f \in \mathrm{LC}_d^0$ with $\mathrm{Q}(f) = \Omega(n^{1-2^{-O(d)}})$. [Childs, Kothari, Kimmel 2012].
- Algorithmic application: the circuit class LC_d^0 can be agnostically PAC learned in time $2^{\tilde{O}(n^{1-2^{-d}})}$. [Kalai, Klivans, Mansour, Servedio 2005]

Implications for Circuit Lower Bounds

- An important frontier problem in circuit complexity is to show that IP2 cannot be computed by $\mathrm{AC}^0 \circ \bigoplus$ circuits of polynomial size.
- Best known lower bound: IP2 cannot be computed by depth-d $\mathrm{AC}^0 \circ \bigoplus$ circuits of size $n^{1+4^{-d}}$[CR96, CGJWX16].
- Our upper bound of $Q(f) = \tilde{O}\left(n^{1-2^{-d}}\right)$ for LC_d^0 circuits f implies that IP2 cannot be computed by depth d $\mathrm{AC}^0 \circ \bigoplus$ circuits of size $O(n^{1+2^{-d}})$, even on a ½+1/poly(n) fraction of inputs. [cf. Tal17].

Open Questions

Is our Upper Bound on $Q(\mathrm{LC}_d^0)$ Tight Up To Logs?

- Recall: we show $Q\left(\mathrm{LC}_d^0\right) = \tilde{O}(n^{1-2^{-d}})$.
- Nearly matches a known lower bound: for every d, there is an LC_d^0 circuit f with $Q(f) = \Omega(n^{1-2^{-O(d)}})$. [Childs, Kothari, Kimmel 2012].

Open Problem: Improve the lower bound to $\boldsymbol{\Omega\left(n^{1-2^{-d}}\right)}$.

Is our Upper Bound on $Q(\mathrm{LC}_d^0)$ Tight Up To Logs?

- Recall: we show $Q\left(\mathrm{LC}_d^0\right) = \tilde{O}(n^{1-2^{-d}})$.
- Nearly matches a known lower bound: for every d, there is an LC_d^0 circuit f with $Q(f) = \Omega(n^{1-2^{-O(d)}})$. [Childs, Kothari, Kimmel 2012].

Open Problem: Improve the lower bound to $\boldsymbol{\Omega\left(n^{1-2^{-d}}\right)}$.

- To accomplish this, it would be enough to achieve the following:

Open Problem: Exhibit a quadratic-size DNF with linear quantum query complexity.

Is our Upper Bound on $Q(\mathrm{LC}_d^0)$ Tight Up To Logs?

- Recall: we show $Q\left(\mathrm{LC}_d^0\right) = \widetilde{O}(n^{1-2^{-d}})$.
- Nearly matches a known lower bound: for every d, there is an LC_d^0 circuit f with $Q(f) = \Omega(n^{1-2^{-O(d)}})$. [Childs, Kothari, Kimmel 2012].

Open Problem: Improve the lower bound to $\boldsymbol{\Omega\left(n^{1-2^{-d}}\right)}$.

- To accomplish this, it would be enough to achieve the following:

Open Problem: Exhibit a quadratic-size DNF with linear quantum query complexity.

- Alternatively, show that every (quadratic? polynomial?) size DNF has sublinear quantum query complexity!

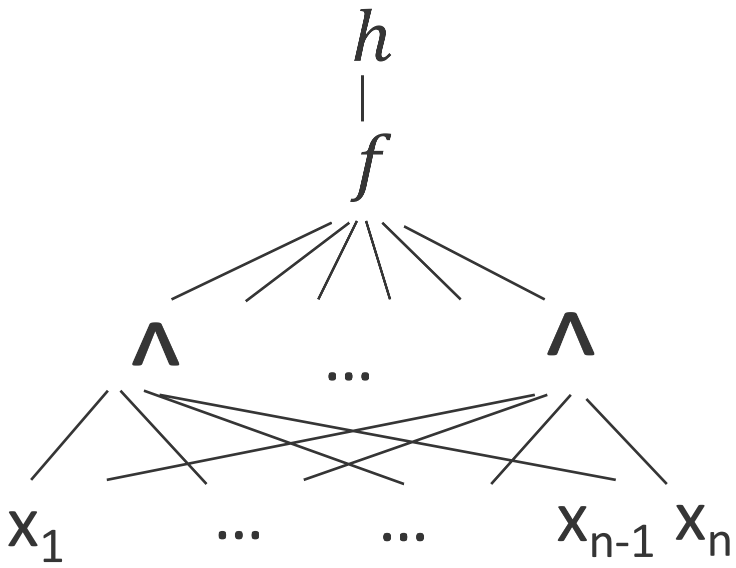
The Lower Bound Argument of [CKK 2012]

- Fact [Beame and Machmouchi 2012, Sherstov 2015]: There is a quadratic size AC^0 circuit C of depth **three**, called SURJECTIVITY, with $Q(C) = \Omega(n)$.
- [CKK 2012] show how to turn C into a **linear-size** circuit C' of depth 3 such that $Q(C') = \Omega(n^{3/4})$.
  - $C' := C_{n^{1/2}} \circ \mathrm{AND}_{n^{1/2}}$.
  - Recall that $Q\left(C_{n^{1/2}} \circ \mathrm{AND}_{n^{1/2}}\right) = \Theta\left(Q\left(C_{n^{1/2}}\right) \cdot Q\left(\mathrm{AND}_{n^{1/2}}\right)\right) = \Omega(n^{3/4})$.

The Lower Bound Argument of [CKK 2012]

- Fact [Beame and Machmouchi 2012, Sherstov 2015]: There is a quadratic size AC^0 circuit C of depth **three**, called SURJECTIVITY, with $Q(C) = \Omega(n)$.
- [CKK 2012] show how to turn C into a **linear-size** circuit C' of depth 3 such that $Q(C') = \Omega(n^{3/4})$.
  - $C' := C_{n^{1/2}} \circ \mathrm{AND}_{n^{1/2}}$.
  - Recall that $Q\left(C_{n^{1/2}} \circ \mathrm{AND}_{n^{1/2}}\right) = \Theta\left(Q\left(C_{n^{1/2}}\right) \cdot Q\left(\mathrm{AND}_{n^{1/2}}\right)\right) = \Omega(n^{3/4})$.
- Then $C'' := C_{n^{1/2}} \circ C'_{n^{1/2}}$ is a linear size circuit of depth 5 such that $Q(C'') = \Omega\left(n^{\frac{7}{8}}\right)$.
- Then $C''' := C_{n^{1/2}} \circ C''_{n^{1/2}}$ is a linear size circuit of depth 7 such that $Q(C''') = \Omega\left(n^{\frac{15}{16}}\right)$. And so on.

Generalizing Our Composition Theorem?



Main Theorem: $Q(h) \leq O\left(\sqrt{Q(f) \cdot n} \cdot polylog(n)\right)$.

- NOTE: if the roles of AND and f are reversed, the composition theorem is **FALSE**.
- But other generalizations of the composition theorem may be possible.

More Classical Implications of Quantum Query Upper Bounds?

- A sublinear upper bound on the quantum query complexity or approximate degree of a circuit class establishes very strong limitations on that class.
  - Immediately implies the class cannot compute IP2, even on average (e.g., [Tal17]).
  - Immediately implies subexponential time agnostic PAC learning algorithms for the class.
  - **Open Question 2: Identify other implications, e.g., Faster SAT algorithms for the class?**

Non-Trivial Query or Communication Upper Bounds for Stronger Circuit Classes?

- Can we show sublinear quantum query or approximate degree upper bounds for circuit classes beyond LC_d^0?
  - Maybe this is a path towards showing IP2 $\notin AC^0 \circ \bigoplus$.
- Alternatively, sublinear **threshold degree** (or subexponential **sign-rank**) upper bounds for a circuit class imply the class cannot compute IP2 in the worst case, and yield PAC learning algorithms for the class.
- Can we show such bounds for interesting circuit classes beyond LC_d^0?
  - E.g., showing LT_2 circuits have sub-exponential sign-rank would mean such circuits cannot compute IP2.